Cepstrum via Homomorphic Filtering

Advanced Digital Signal Processing

Christos Samaras

1 Voice samples acquisition

This study aims to identify qualitative and quantitative differences between male and female vowels and how these are reflected in the cepstrum domain. Vowel sounds "a", "e", "o", "i", "u" from a male participant and a female participant are recorded. Recording lasted 2 seconds, with 8000Hz sampling rate, and was performed using MATLAB in a "Dell Inspiron 3593 i7" laptop. 20 ms of each vowel signal are presented below.

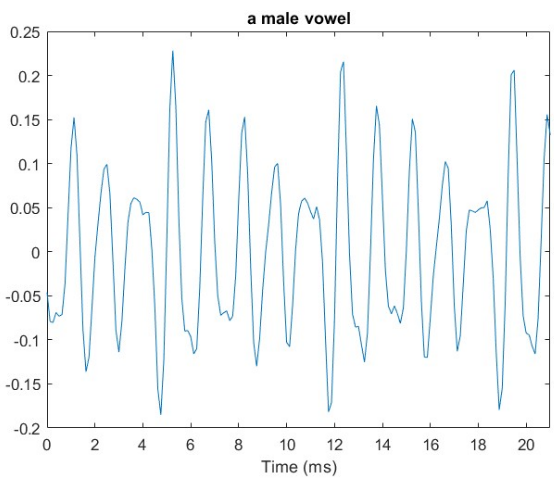

Figure 1: Male vowel signal of "a"

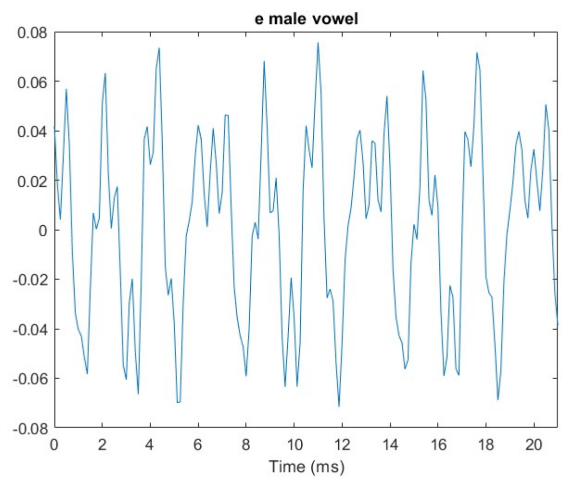

Figure 2: Male vowel signal of "e"

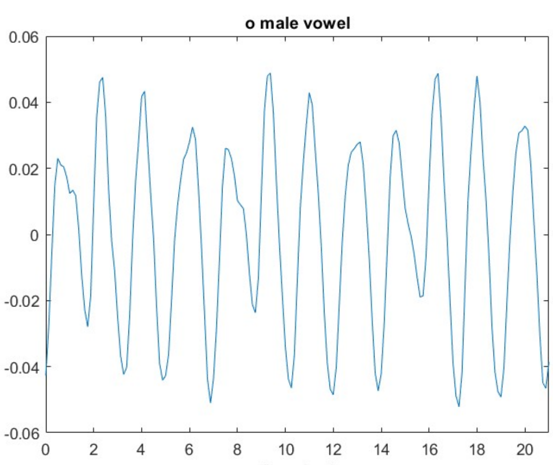

Figure 3: Male vowel signal of "o"

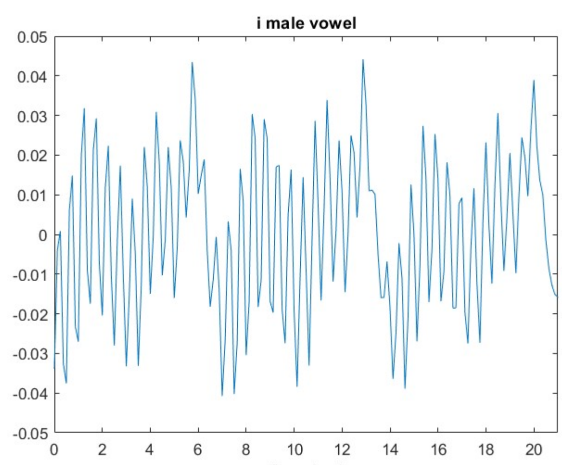

Figure 4: Male vowel signal of "i"

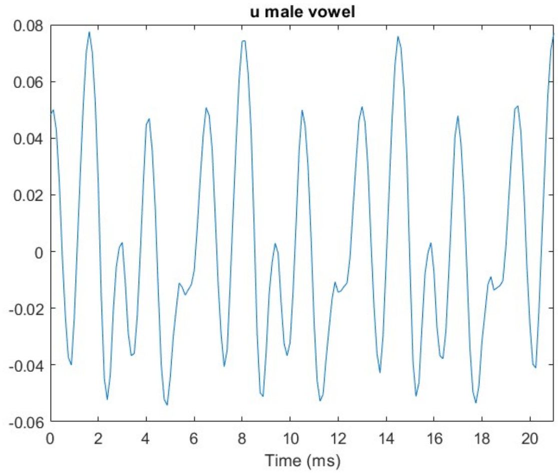

Figure 5: Male vowel signal of "u"

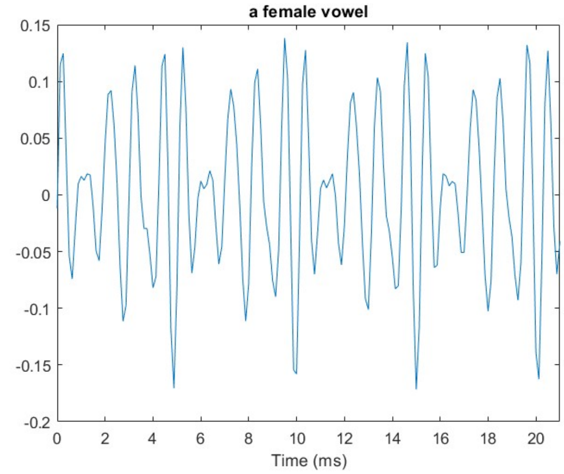

Figure 6: Female vowel signal of "a"

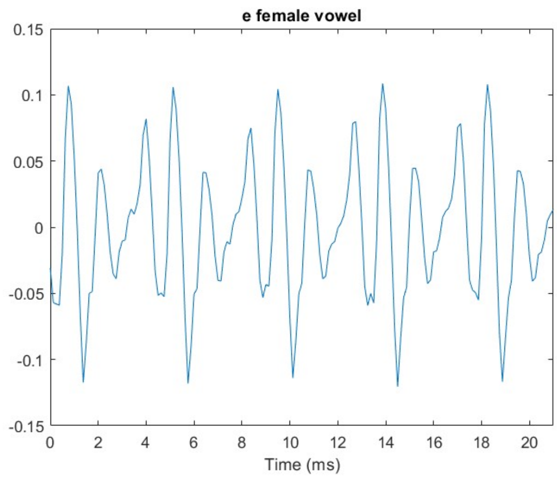

Figure 7: Female vowel signal of "e"

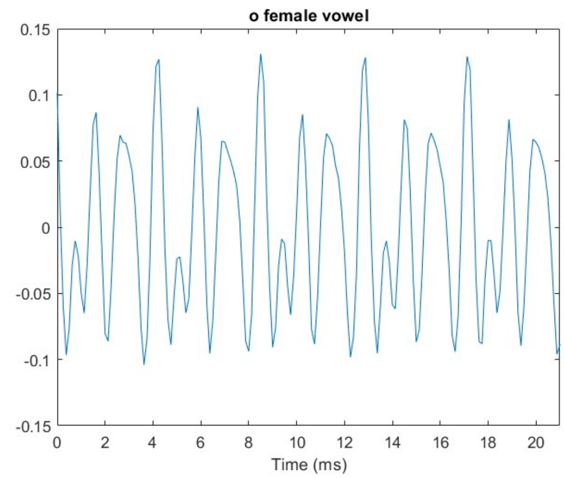

Figure 8: Female vowel signal of "o"

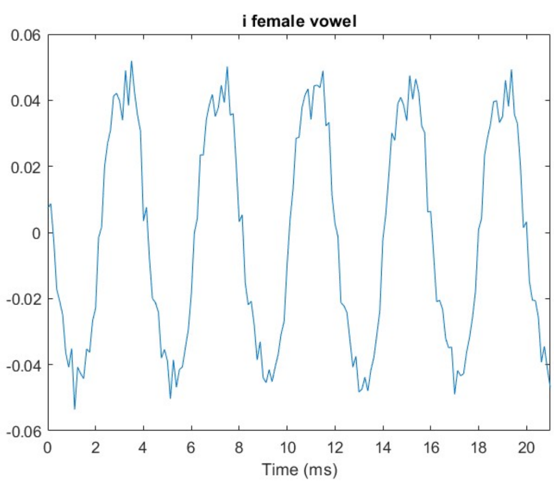

Figure 9: Female vowel signal of "i"

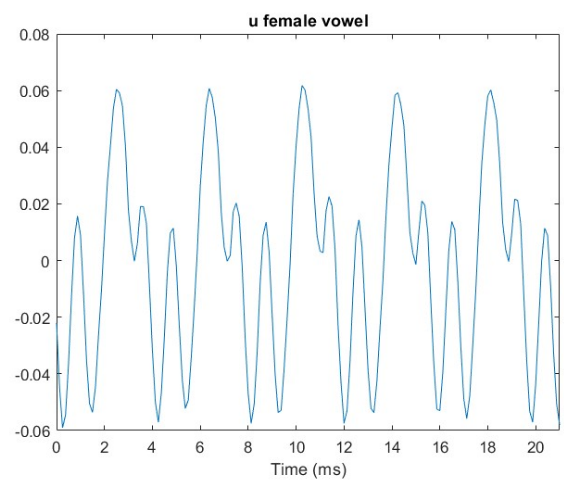

Figure 10: Female vowel signal of "u"

2 Cepstrum Domain

Real and Complex Cepstrum of a sequence x[n] are defined as

$$c_{\text{real}}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log[|X(\omega)|]e^{j\omega n}\, d\omega$$

$$c_{\text{complex}}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log[X(\omega)]e^{j\omega n}\, d\omega$$

$$where\ X(\omega)\ denotes\ the\ Fourier\ Transform\ of\ a\ sequence\ x[n]$$

A periodic part of each vowel is extracted. Then, Hamming Window is applied on it. Real and Complex Cepstrum are calculated both for original signals and for windowed signals. For this purpose, MATLAB functions "rceps" and "cceps" are used for real and complex cepstrum calculation respectively. Their plots are displayed below.

A general observation is that peaks on quefrency (rahmonics) are more clearly depicted in Real Cepstrum than Complex Cepstrum. Additionally, female rahmonics appear ealrier in quefrency and therefore decay faster compared to male rahmonics. This phenomenon indicates that female pitch period is shorter than male pitch period and consequently female fundamental frequency is greater than male fundamental frequency. That's why female voice has generally more peaks than male voice in cepstral domain (see also "3 Voice, a convolution of 2 signals").

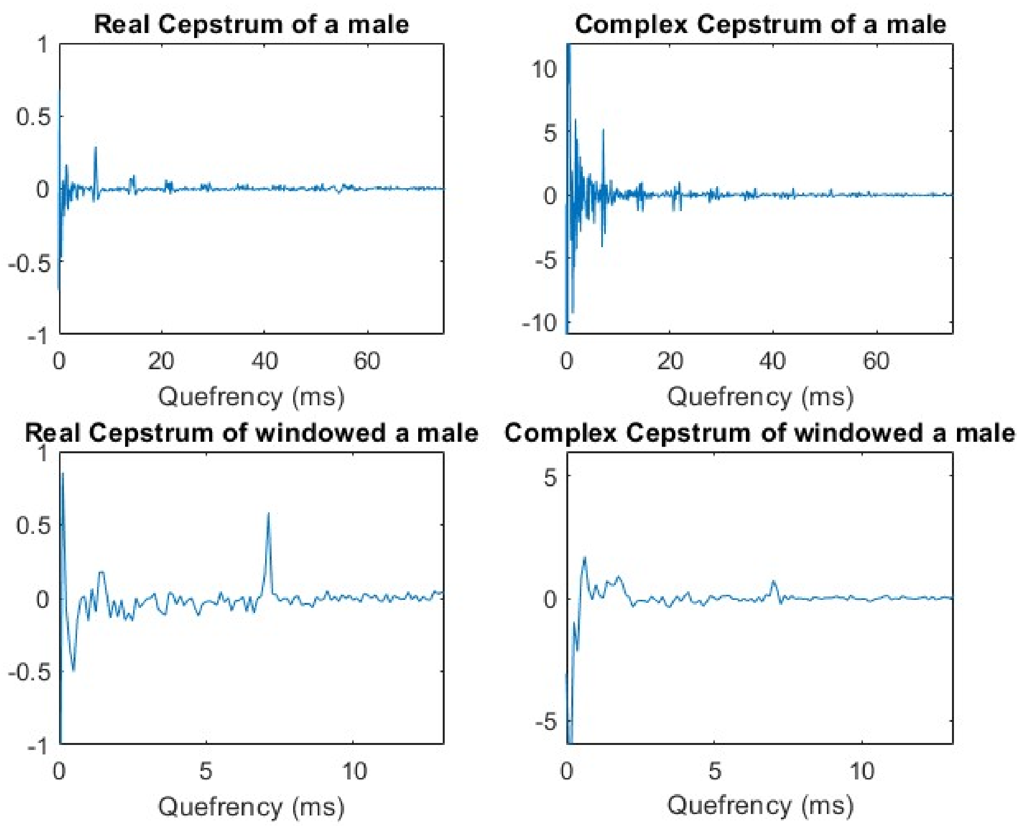

Figure 11: Real and Complex Cepstrum of "a" male vowel

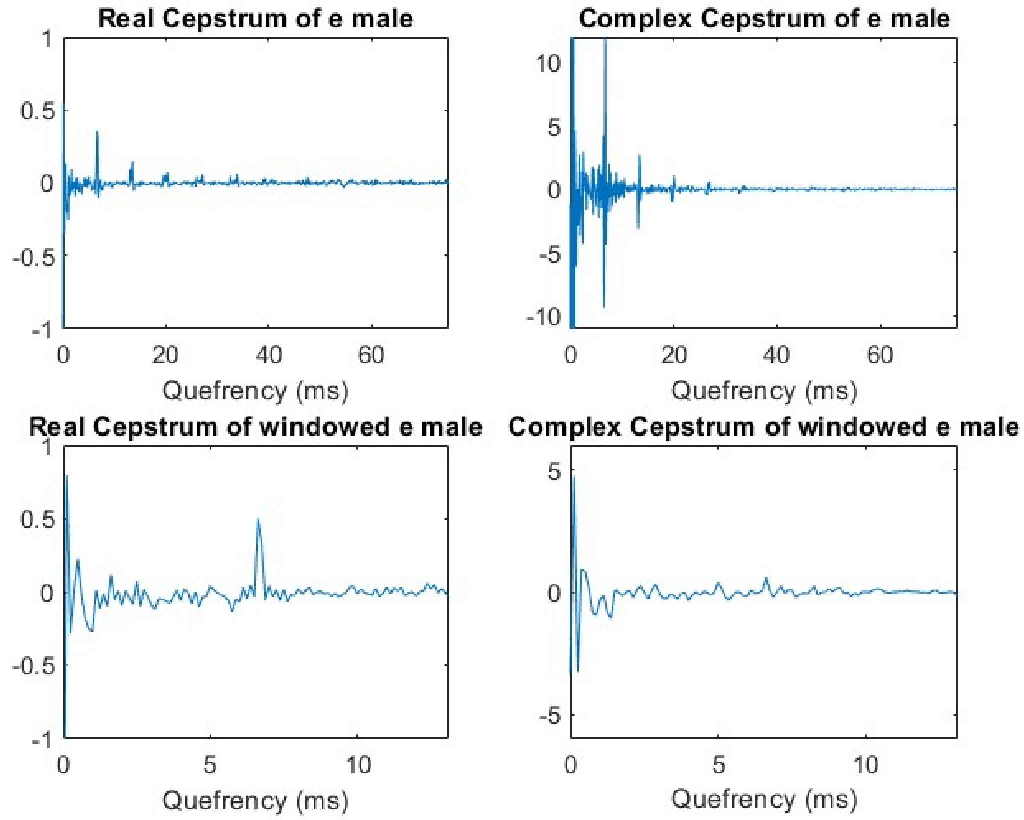

Figure 12: Real and Complex Cepstrum of ”e” male vowel

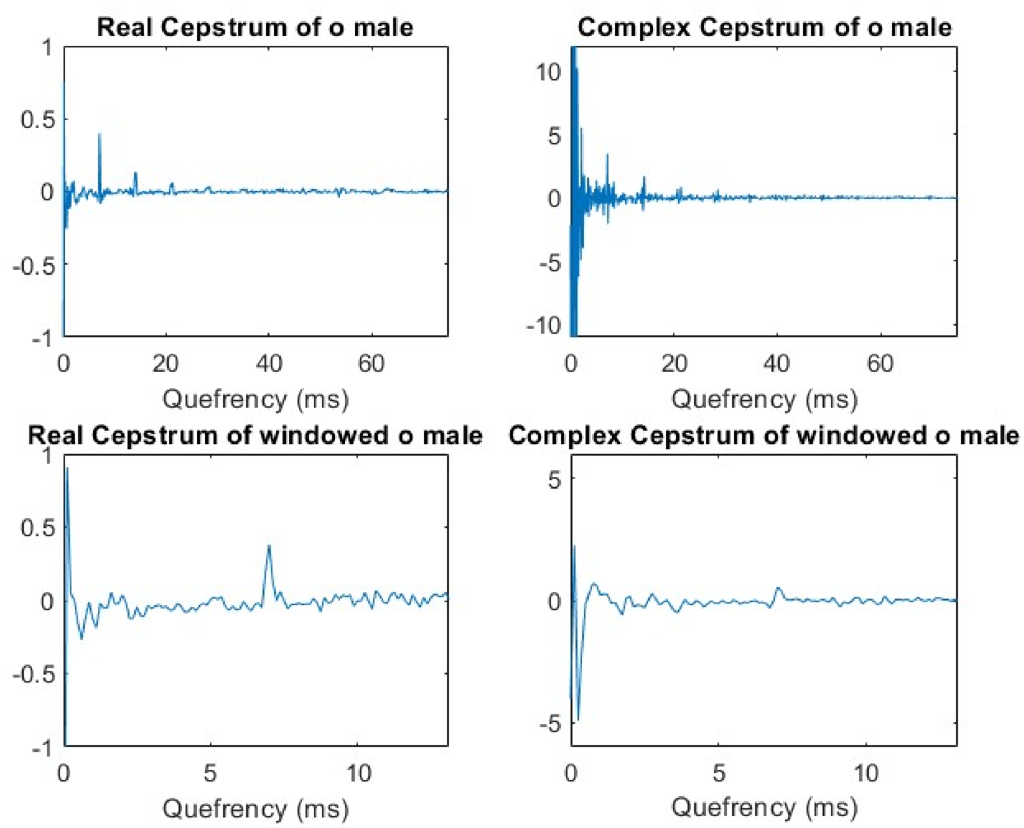

Figure 13: Real and Complex Cepstrum of ”o” male vowel

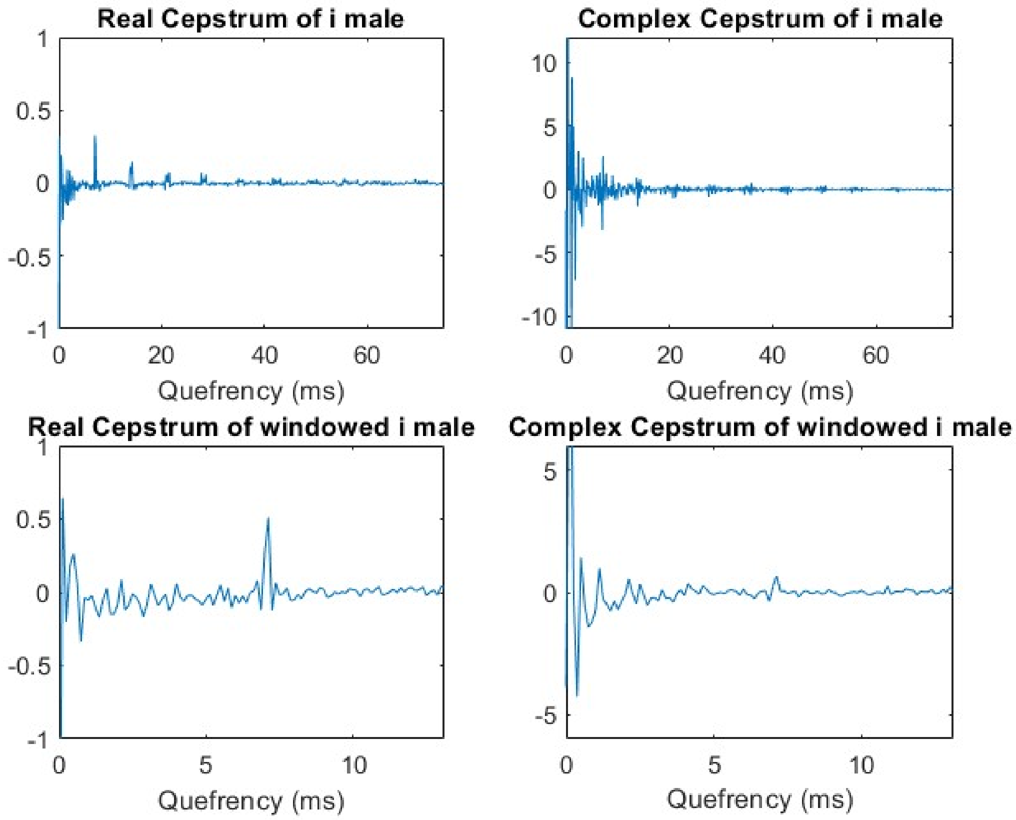

Figure 14: Real and Complex Cepstrum of "i" male vowel

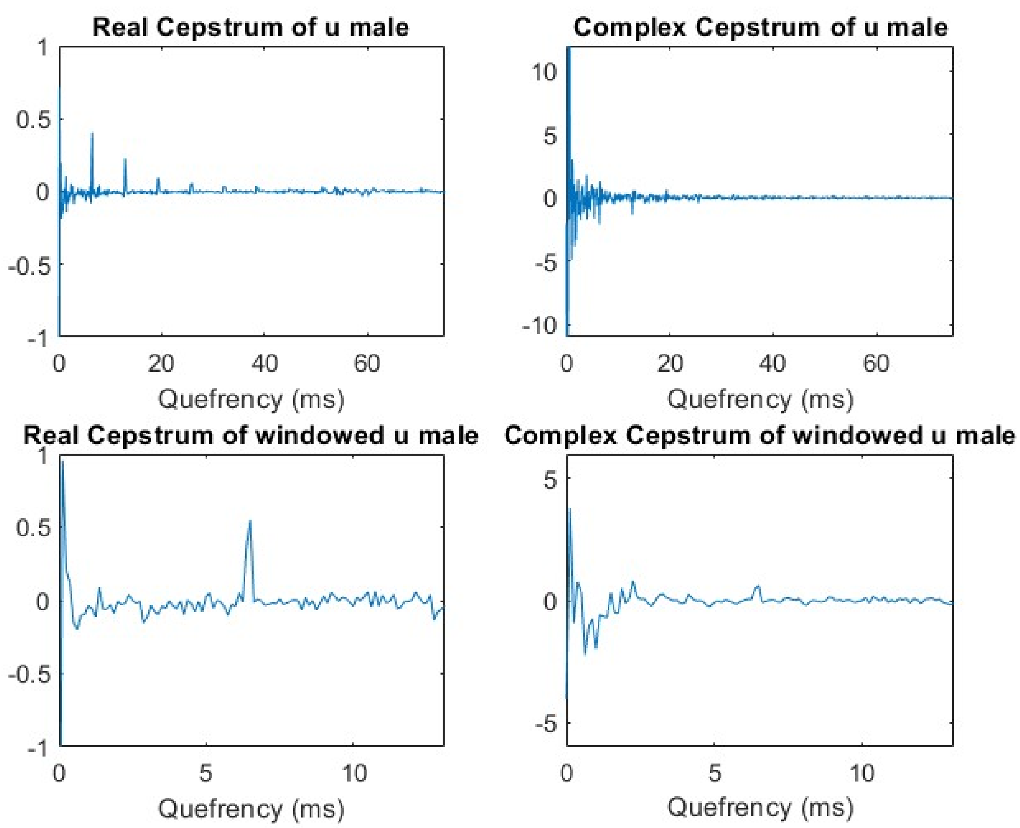

Figure 15: Real and Complex Cepstrum of "u" male vowel

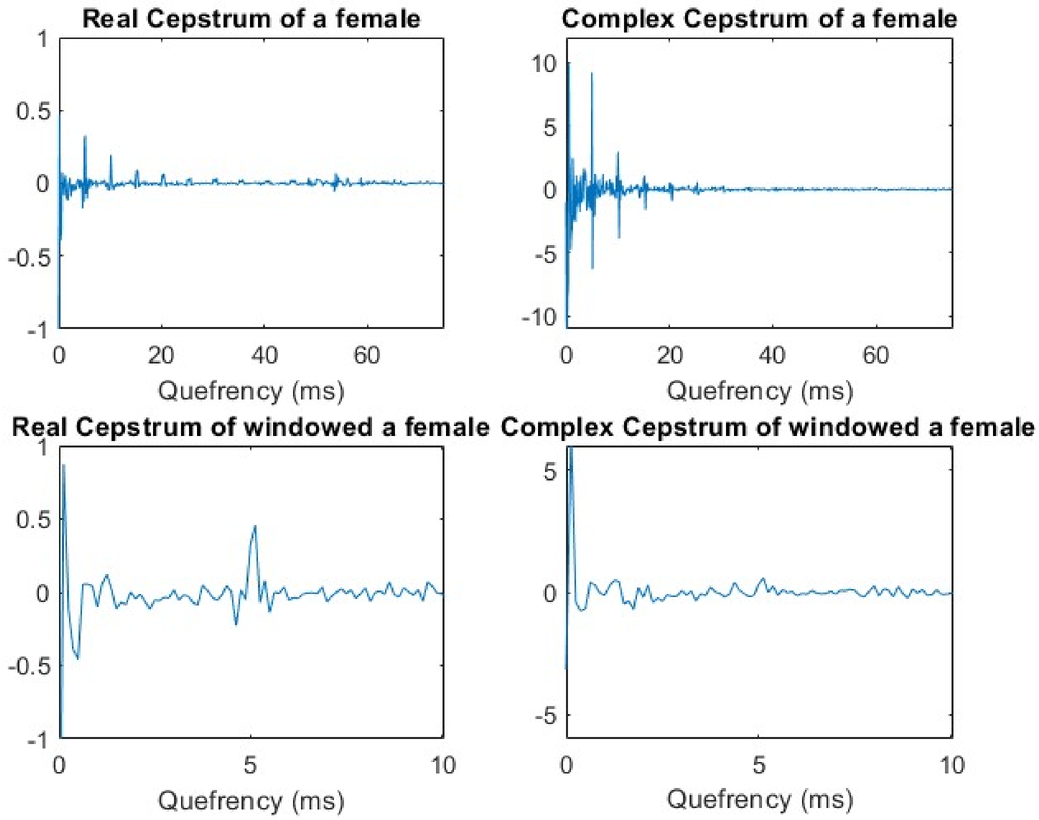

Figure 16: Real and Complex Cepstrum of ”a” female vowel

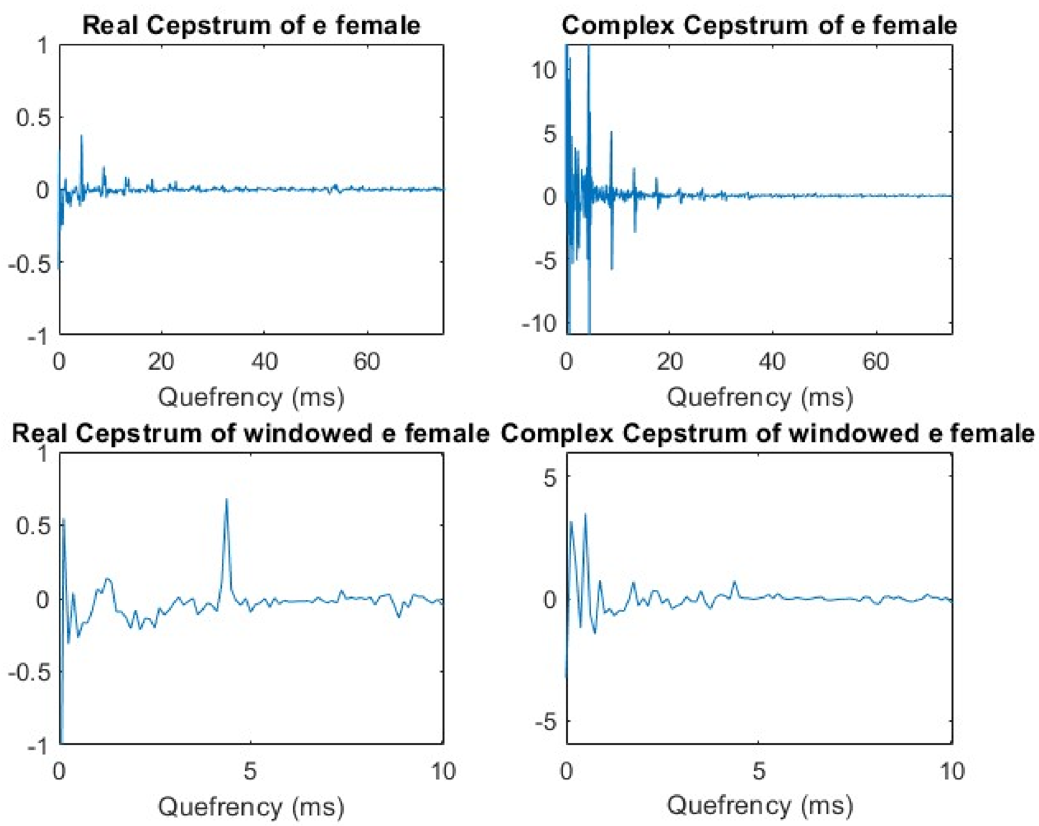

Figure 17: Real and Complex Cepstrum of ”e” female vowel

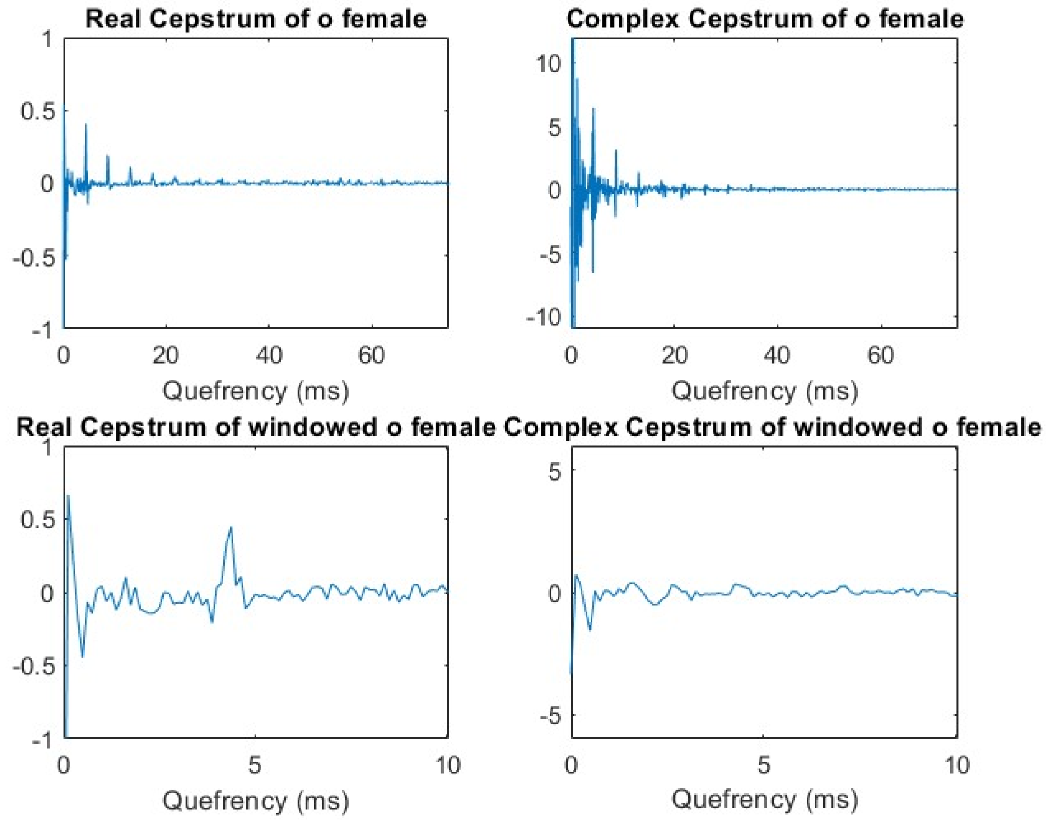

Figure 18: Real and Complex Cepstrum of ”o” female vowel

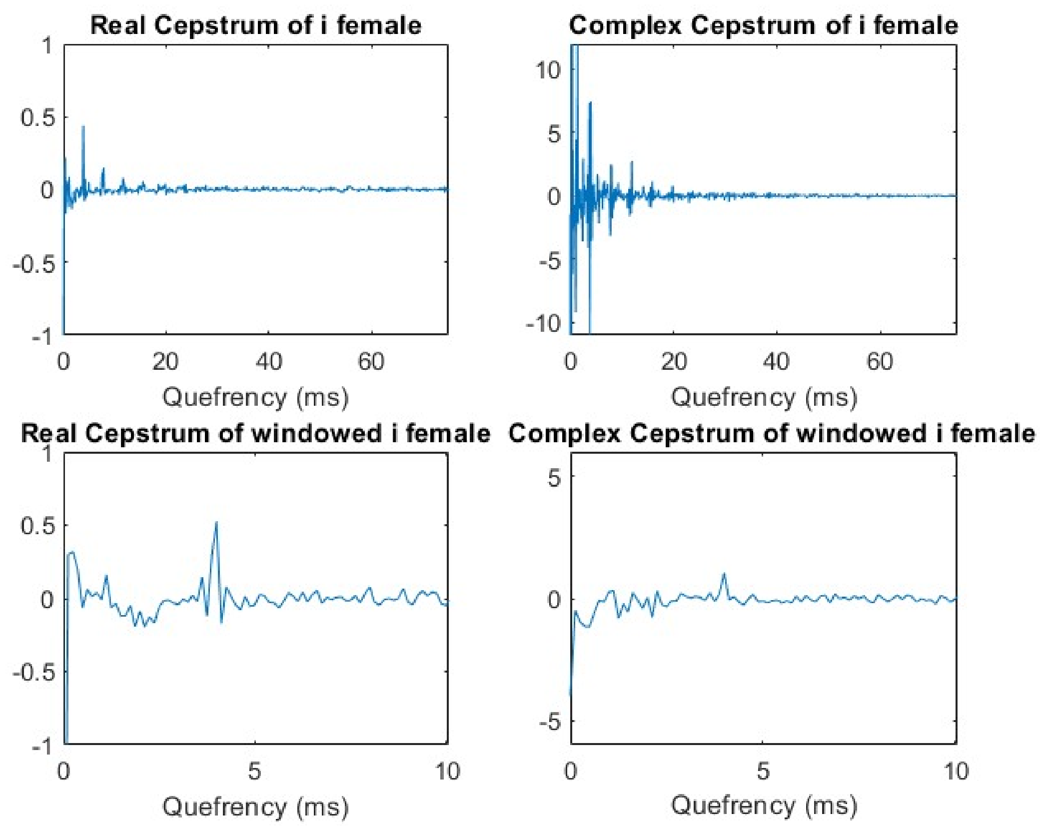

Figure 19: Real and Complex Cepstrum of ”i” female vowel

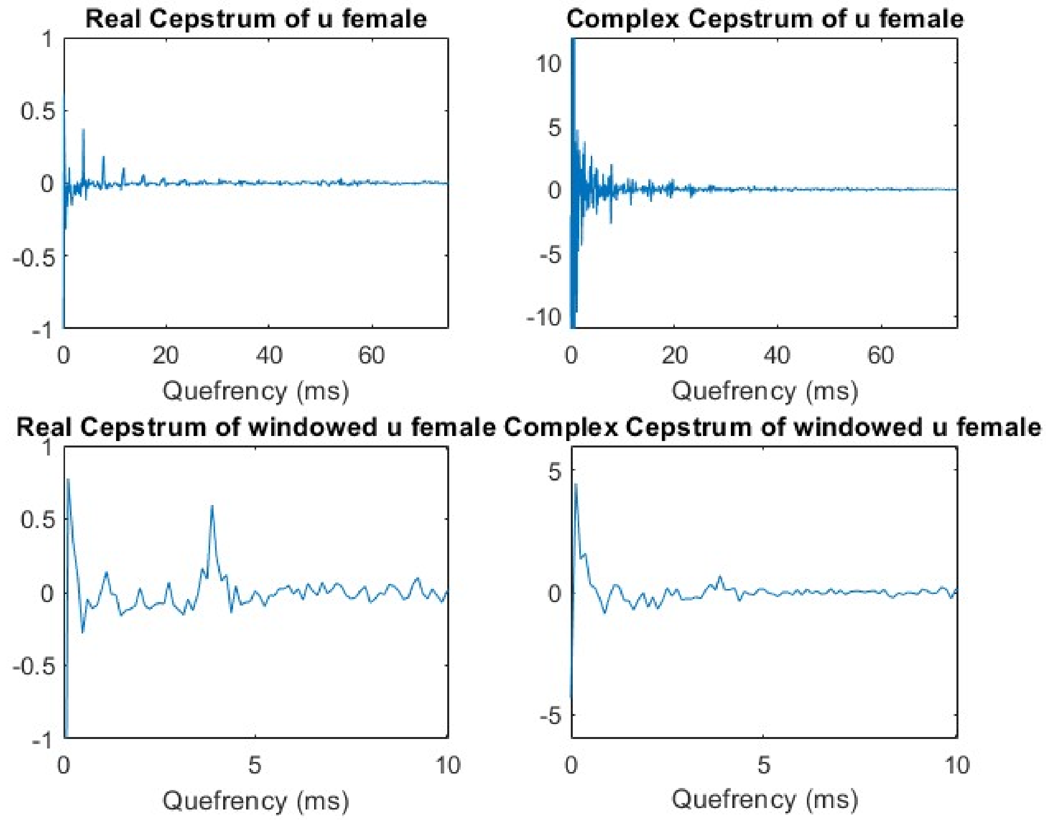

Figure 20: Real and Complex Cepstrum of "u" female vowel

3 Voice, a convolution of 2 signals

Voice can be considered as the convolution of vocal tract impulse response h[n] with vocal cords pulses p[n]. Air flow exhaled from lungs induces self-oscillations of the vocal cords. These oscillations generate acoustic waves that propagate through vocal tract and are emitted outwards. As the last one varies, e.g. different tongue positions, different sounds are perceived. Pitch is determined by vocal cords oscillations frequency. This frequency is influenced by tension and mass of folds. In the right side, human vocal tract is presented.

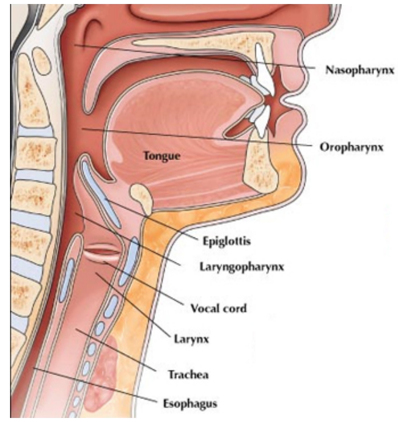

Figure 21: Vocal Tract

Sequences h[n] and p[n] are exported through vowels' complex cepstrum. For each vowel, a periodic part of it is taken. Then, Hamming Window is applied. Theoretically, window's length should be 2-3 times of pitch periods. In fact, length is set at least 4 pitch periods in order to achieve more realistic results. Complex cepstrum of windowed signal is computed and two different lifters are applied, a low-pass lifter for h[n] acquisition and a high-pass lifter for p[n] acquisition. Lifter length is set appropriately for each sample. For low-pass lifter, length is set equal with approximately half pitch period. High-pass lifter extended from approximately first pitch to half of cepstrum length . The exact limits of each lifter are determined by careful observation of each signal and its cepstrum.

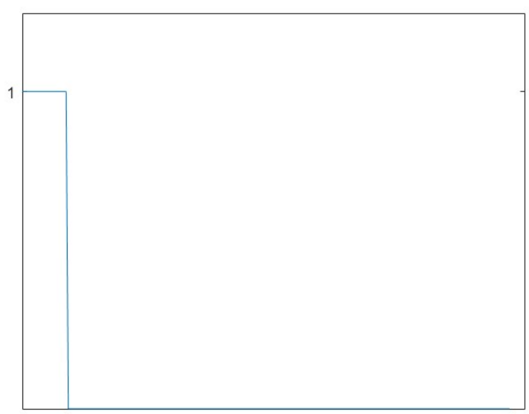

Figure 22: Low-Pass Lifter

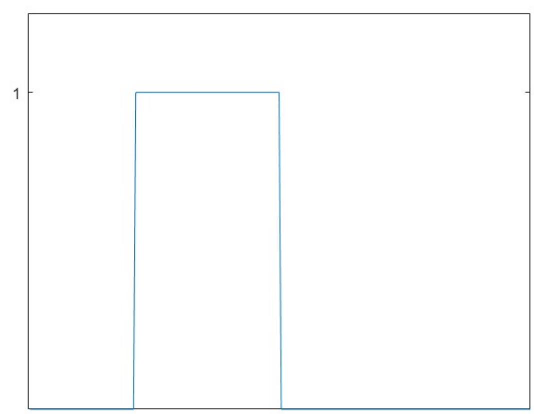

Figure 23: High-Pass Lifter

After, inverse of cepstrum is calculated via "icceps" MATLAB function and h[n], p[n] sequences for each male and female vowel are extracted and plotted below. Mainly, male vocal tract impulse response has greater amplitude compared to female, which is likely due to its larger size compared to female. In contrast, frequency of female signals is higher compared to male signals. This can be explained by the fact that women generally have shorter and thinner vocal cords than men. Since shorter and thinner vocal folds vibrate faster, they produce a higher fundamental frequency. So, pitch is connected to vocal folds and that's why vowels of each participant have approximately equal pitch period. Thus, vocal tract geometry is what finally defines which vowel is heard.

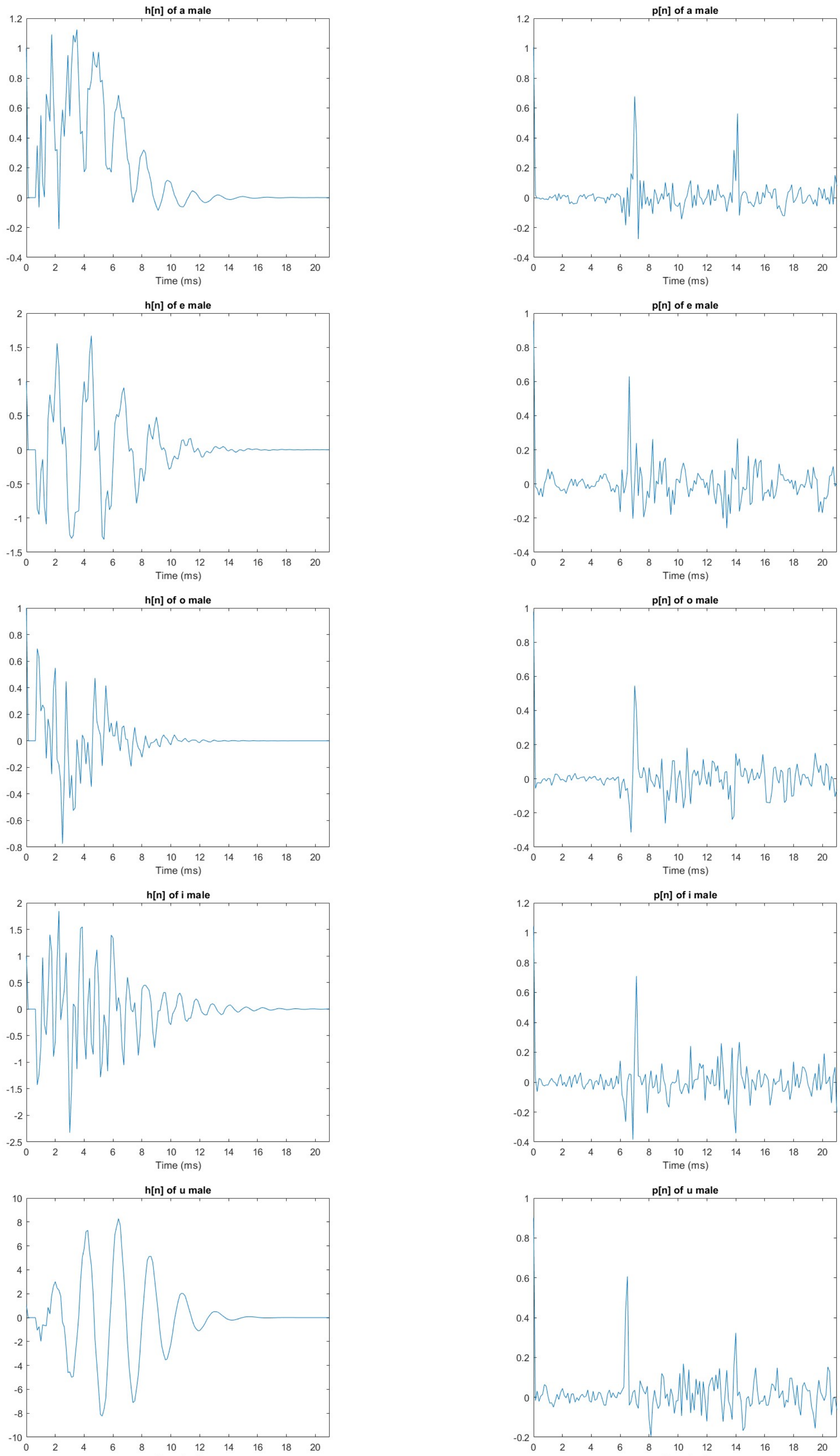

Figure 24: h[n] and p[n] of male vowels

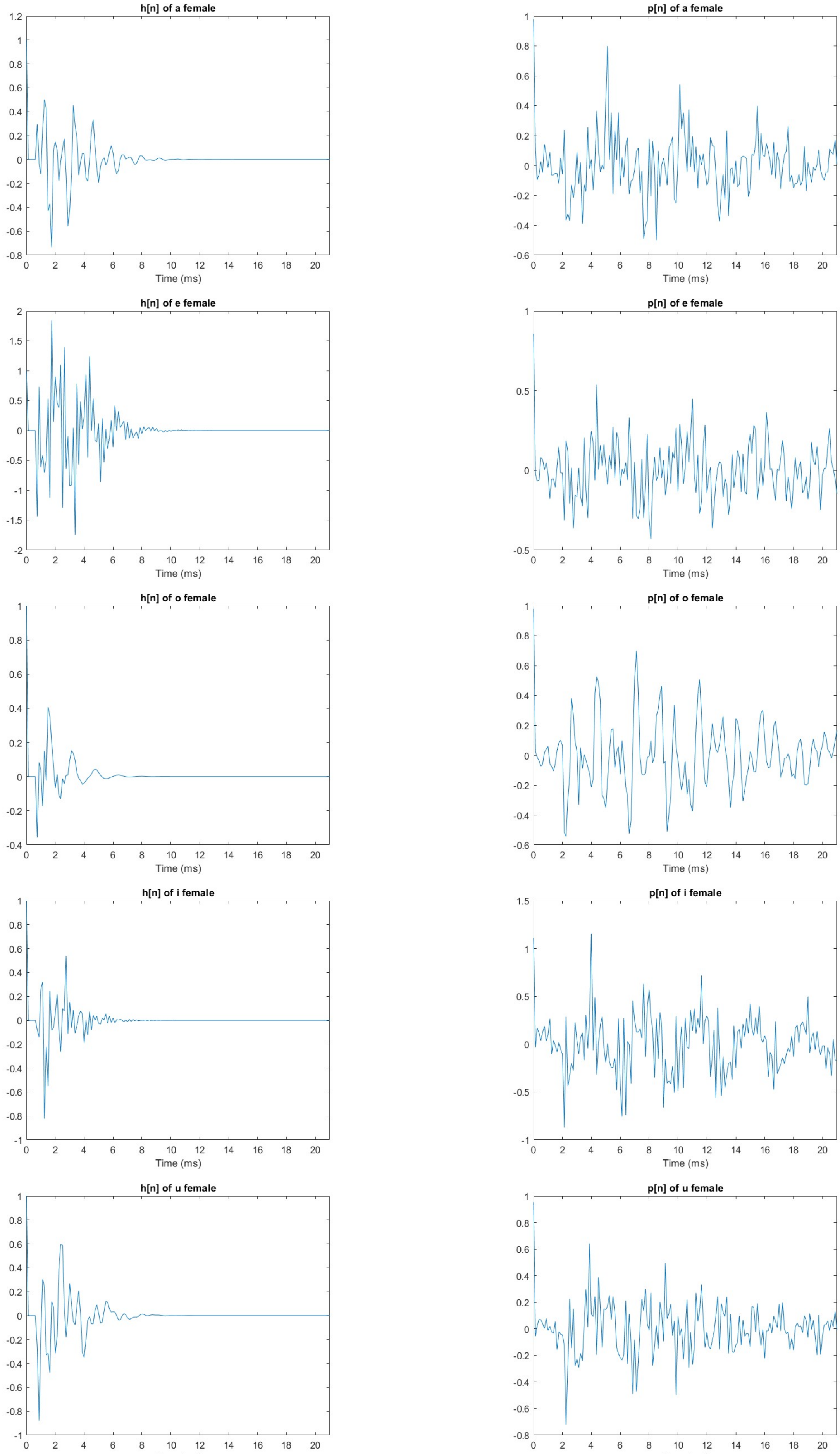

Figure 25: h[n] and p[n] of female vowels

4 Vowel re-synthesize

Finally, an attempt was made to reconstruct the original vowels from the sequences h[n] and p[n] that were previously extracted. For this purpose, the path shown in the diagram below was followed.

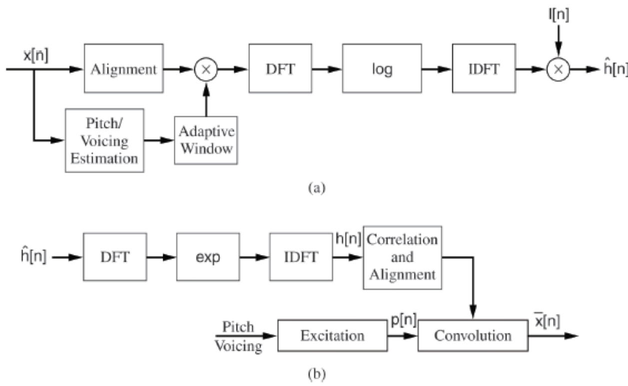

Figure 26: Re-synthesize system

Reconstructed vowels have not the same quality with the original ones. They are sounded like muffled and a little bit buzzy. Vowel "a" of male and female is the most qualitative reconstructed signal.

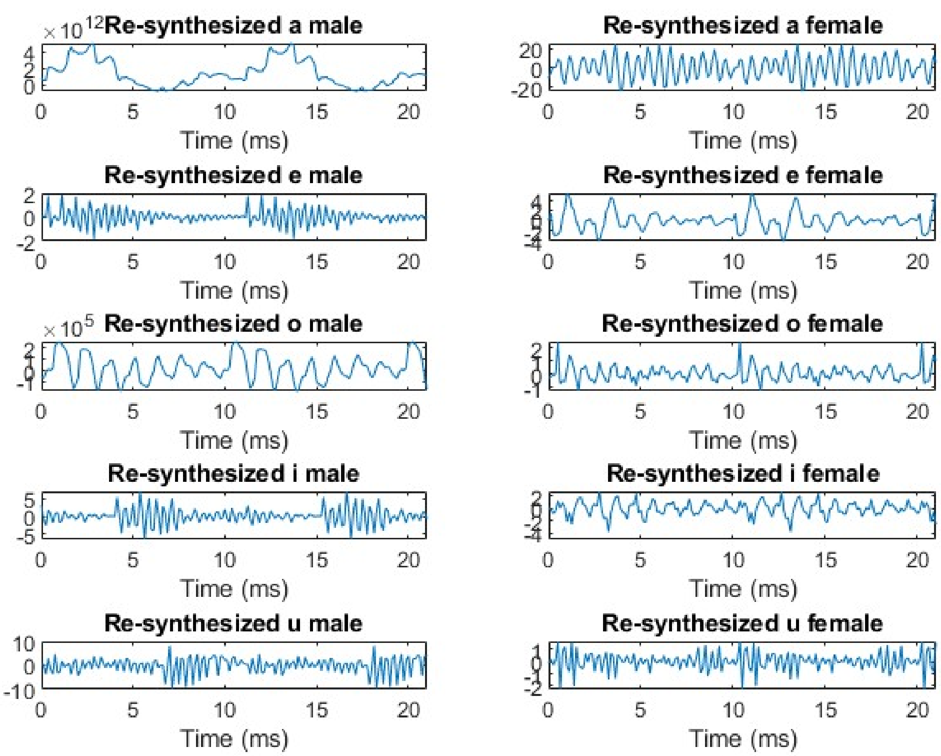

Figure 27: Re-synthesized vowels

5 MATLAB code

Voice samples acquired with "VoiceRecording.m" file. Here is its code:

```
%% Voice Samples Recording
fs=8000;
s=audiorecorder(fs, 16, 1);
recordblocking(s, 2);
vsam=getaudiodata(s); %change it for each vowel
    saveDir = '...'; %Save address
    if ~exist(saveDir, 'dir')
      mkdir(saveDir);
    end
    filename = fullfile(saveDir, 'a_male.mat'); %change it for each vowel
    save(filename, 'vsam', 'fs') %change it for each vowel

%% Cepstrum plot
% After you can load mat files and plot cepstrums with the following code
% doing the proper changes
figure(1)
subplot(3,1,1);
plot(vsam)   %change it for each vowel
title('"a" male voice sample time domain')  %change it for each vowel
subplot(3,1,2);
plot(rceps(vsam))   %change it for each vowel
xlim([-0.001 600]);
ylim([-1 1])
title('Real Cepstrum of "a" male voice sample')  %change it for each vowel
subplot(3,1,3);
plot(cceps(vsam))    %change it for each vowel
xlim([-0.001 1100]);
ylim([-11 12])
title('Complex Cepstrum of "a" male voice sample')%change it for each vowel

%%          Voice Samples
% a_male = vsam  a_female = vsaf
% e_male = vsem  e_female = vsef
% o_male = vsom  o_female = vsof
% i_male = vsim  i_female = vsif
% u_male = vsum  u_female = vsuf
```

Study was implemented in "exercise4.mlx" file, Here is its code:

```
% ---------------------------------------------------------------------
%                              exercise 4
% ---------------------------------------------------------------------
addpath("C:\Users\samag\OneDrive\ADSP\exercise4\Voice Samples)
files={'a_male.mat', 'e_male.mat', 'o_male.mat', 'i_male.mat', 'u_male.mat', 'a_female.mat', 'e_female.mat', 'o_female.mat', 'i_female.mat', 'u_female.mat'};
for k=1:length(files)
    load(files{k});            % loading .mat file of each vowel
end

fs=8000;
Nall=16000;
tall=(0:Nall-1)/fs;

om = 4;                        % Multiply factor for male signal window
Nmale = 57;                    % Male vowels are periodic each N=57 samples on average
mwhamm = hamming(om*Nmale);    % Hamming Window for male vowels
ssm = 6001;                    % Male signal start
sem = 6000+om*Nmale;           % Male signal end
of = 5;                        % Multiply factor for female signal window
Nfemale = 35;                  % female vowels are periodic each Nfemale=40 samples on average
fwhamm = hamming(of*Nfemale);% Hamming Window for woman
ssf = 6001;                    % signal start
sef = 6000+of*Nfemale;         % signal end

Nsm = (sem-ssm+1);             % N for original male signal
tsm = ((0:Nsm-1)/fs)*1000;     % Time vector for male
Nsf = (sef-ssf+1);             % N for original female signal
tsf = ((0:Nsf-1)/fs)*1000;     % Time vector for female


malevowels={vsam, vsem, vsom, vsim, vsum};  % Vowel signals for male
malevowelstitle={'a male', 'e male', 'o male', 'i male', 'u male'};
mwcc = cell(1, 5);                          % Store complex cepstrums of male windowed signals
mwrc = cell(1, 5);                          % Store real cepstrums of male windowed signals

femalevowels={vsaf, vsef, vsof, vsif, vsuf};% Vowel signals for male
femalevowelstitle={'a female', 'e female', 'o female', 'i female', 'u female'};
fwcc = cell(1, 5);                          % Store complex cepstrums of fmale windowed signals
fwrc = cell(1, 5);                          % Store real cepstrums of fmale windowed signals

fprintf('------------------------------- Task 2 --------------------------------')
for i=1:5
    mwvs = malevowels{i}(ssm:sem).*mwhamm;  % male windowed voice sample
    mwcc{i} = cceps(mwvs);                  % Calculate complex cepstrum
    mwrc{i} = rceps(mwvs);                  % Calculate real cepstrum

    % Plot time-domain signals
    figure;
    plot(tsm, malevowels{i}(ssm:sem))
    xlim([0 21])
    xlabel('Time (ms)')
    title(malevowelstitle{i})
```

```

    figure;
    subplot(2, 2, 1)
    plot(1000*tall, rceps(malevowels{i}))
    xlim([-0.0001 75]);
    ylim([-1 1])
    xlabel('Quefrency (ms)')
    title(['Real Cepstrum of ' malevowelstitle{i}])

    subplot(2, 2, 3)
    plot(tsm, rceps(malevowels{i}(ssm:sem).*mwhamm))
    xlim([-0.0001 length(tsm)/17.4])
    ylim([-1 1])
    xlabel('Quefrency (ms)')
    title(['Real Cepstrum of windowed ' malevowelstitle{i}])

    subplot(2, 2, 2)
    plot(1000*tall, cceps(malevowels{i}))
    xlim([-0.0001 75])
    ylim([-11 12])
    xlabel('Quefrency (ms)')
    title(['Complex Cepstrum of ' malevowelstitle{i}])

    subplot(2, 2, 4)
    plot(tsm, cceps(malevowels{i}(ssm:sem).*mwhamm))
    xlim([-0.0001 length(tsm)/17.4])
    ylim([-6 6])
    xlabel('Quefrency (ms)')
    title(['Complex Cepstrum of windowed ' malevowelstitle{i}])
fprintf('-----------------------------------------------------------')
    fwvs = femalevowels{i}(ssf:sef).*fwhamm;  % male windowed voice sample
    fwcc{i} = cceps(fwvs);                    % Calculate complex cepstrum
    fwrc{i} = rceps(fwvs);                    % Calculate real cepstrum

    % Plot time-domain signals
    figure;
    plot(tsf, femalevowels{i}(ssf:sef))
    xlim([0 21])
    xlabel('Time (ms)')
    title(femalevowelstitle{i})

    figure;
    subplot(2, 2, 1)
    plot(1000*tall, rceps(femalevowels{i}))
    xlim([-0.0001 75]);
    ylim([-1 1])
    xlabel('Quefrency (ms)')
    title(['Real Cepstrum of ' femalevowelstitle{i}])

    subplot(2, 2, 3)
    plot(tsf, rceps(femalevowels{i}(ssf:sef).*fwhamm))
    xlim([-0.0001 length(tsf)/17.4])
    ylim([-1 1])
    xlabel('Quefrency (ms)')
    title(['Real Cepstrum of windowed ' femalevowelstitle{i}])

    subplot(2, 2, 2)
    plot(1000*tall, cceps(femalevowels{i}))
    xlim([-0.0001 75])
    ylim([-11 12])
```

```
    xlabel('Quefrency (ms)')
    title(['Complex Cepstrum of ' femalevowelstitle{i}])

    subplot(2, 2, 4)
    plot(tsf, cceps(femalevowels{i}(ssf:sef).*fwhamm))
    xlim([-0.0001 length(tsf)/17.4])
    ylim([-6 6])
    xlabel('Quefrency (ms)')
    title(['Complex Cepstrum of windowed ' femalevowelstitle{i}])
fprintf('-----------------------------------------------------------------')
end

% Liftering parameters for h and p sequences
hmlowerlim = 7;
hmupperlim = 27;
hmwind = [zeros(1, hmlowerlim-1), ones(1, hmupperlim-hmlowerlim+1), zeros(1, length(
    mwcc{1}) - hmupperlim)];
pmwind = [zeros(1, 48), ones(1, floor(length(mwcc{1})/2)-48), zeros(1, length(mwcc{1})
    -floor(length(mwcc{1})/2))];

hm = cell(1, 5);            % Store h[n] male sequences
pm = cell(1, 5);            % Store p[n] male sequences

hflowerlim=7;
hfupperlim=15;
hfwind = [zeros(1, hflowerlim-1), ones(1, hfupperlim-hflowerlim+1), zeros(1, length(
    fwcc{1})-hfupperlim)];                                % lifter window for h sequence
pfind = [zeros(1,hfupperlim+2), ones(1, floor(length(fwcc{1})/2)-(hfupperlim+2)),
    zeros(1, length(fwcc{1})-floor(length(fwcc{1})/2))]; % lifter window for p
    sequence

hf = cell(1, 5);            % Store h[n] female sequences
pf = cell(1, 5);            % Store p[n] female sequences

fprintf('------------------------------- Task 3 -------------------------------')
for i = 1:length(mwcc)
    hmcc = mwcc{i} .* hmwind';
    pmcc = mwcc{i} .* pmwind';
    hm{i} = icceps(hmcc(hmlowerlim:hmupperlim));
    pm{i} = icceps(pmcc(48:floor(length(mwcc{i})/2)));

    figure;
    plot(tsm, icceps(hmcc)) % male h[n]
    xlabel('Time (ms)')
    xlim([0 21])
    title(['h[n] of ' malevowelstitle{i}])
    figure;
    plot(tsm, icceps(pmcc)) % male p[n]
    xlabel('Time (ms)')
    xlim([0 21])
    title(['p[n] of ' malevowelstitle{i}])
fprintf('-----------------------------------------------------------------')
end

for i = 1:length(fwcc)
    hfcc = fwcc{i} .* hfwind';
    pfcc = fwcc{i} .* pfind';
    hf{i} = icceps(hfcc(hmlowerlim:hmupperlim));
    pf{i} = icceps(pfcc(48:floor(length(fwcc{i})/2)));

```

```
    figure;
    plot(tsf, icceps(hfcc)) % female h[n]
    xlabel('Time (ms)')
    xlim([0 21])
    title(['h[n] of ' femalevowelstitle{i}])
    figure;
    plot(tsf, icceps(pfcc)) % female p[n]
    xlabel('Time (ms)')
    xlim([0 21])
    title(['p[n] of ' femalevowelstitle{i}])
fprintf('------------------------------------------------------------------')
end

fprintf('-------------------------------- Task 4 --------------------------------')
% Reconstruction of vowel signals
duration = 3;                           % Duration in seconds for reconstructed signal
total_samples = duration * fs;
silence_duration = 0.5;                 % Duration of silence in seconds
silence_length = silence_duration * fs; % Length in samples
silence = zeros(silence_length, 1);     % quiet time among reconstructed signals audio play
resynmvsRecurr = cell(1, 5);            % To store each resynthesized male vowel
resynfvsRecurr = cell(1, 5);            % To store each resynthesized female vowel

for i = 1:length(hm)
    Hm = fft(hm{i});
    new_hm = ifft(exp(Hm));

    % Cross-correlation and alignment
    [aligned_hm, lags] = xcorr(new_hm, hm{i});
    [~, max_index] = max(aligned_hm);
    shift = lags(max_index);
    if shift > 0
        hm_aligned = new_hm(shift:end);
    elseif shift < 0
        hm_aligned = [zeros(abs(shift), 1); new_hm];
    else
        hm_aligned = new_hm;
    end

    resynmvs = conv(pm{i}, hm_aligned);                % male re-synthesized vowel
    Nrepeat = ceil(total_samples / length(resynmvs));
    resynmvsRecurr{i} = repmat(resynmvs, Nrepeat, 1); % recurrence of male re-synthesized vowel
end

for i = 1:length(hf)
    H = fft(hf{i});
    new_h = ifft(exp(H));

    % Cross-correlation and alignment
    [aligned_h, lags] = xcorr(new_h, hf{i});
    [~, max_index] = max(aligned_h);
    shift = lags(max_index);
    if shift > 0
        h_aligned = new_h(shift:end);
    elseif shift < 0
        h_aligned = [zeros(abs(shift), 1); new_h];
    else
        h_aligned = new_h;
```

```
    end

    resynfvs = conv(pf{i}, h_aligned);                % female re-synthesized vowel
    Nrepeat = ceil(total_samples / length(resynfvs));
    resynfvsRecurr{i} = repmat(resynfvs, Nrepeat, 1); % recurrence of female re-
       synthesized vowel
end

for i=1:5
    figure;
    plot(tsm, resynmvsRecurr{i}(ssm:sem))
    xlim([0 21])
    title(['Re-synthesized ' malevowelstitle{i}])
fprintf('----------------------------------------------------------------')
    figure;
    plot(tsf, resynfvsRecurr{i}(ssf:sef))
    xlim([0 21])
    title(['Re-synthesized ' femalevowelstitle{i}])
fprintf('----------------------------------------------------------------')
end

mResynTest=[resynmvsRecurr{1}; silence; resynmvsRecurr{2}; silence; resynmvsRecurr{3};
    silence; resynmvsRecurr{4}; silence; resynmvsRecurr{5}; silence]; % male audio
fResynTest=[resynfvsRecurr{1}; silence; resynfvsRecurr{2}; silence; resynfvsRecurr{3};
    silence; resynfvsRecurr{4}; silence; resynfvsRecurr{5}; silence]; % female audio
% Resynthezise test for male: Vowels should be heard with this line -> a, e, o, i, u
% !! Delete the following 2 "%" to hear resynthesized vowels !!
%sound(mResynTest, fs)
%sound(fResynTest, fs)
```